ALCATEL WEB TOUCH ONE

Hardware Design Information

Version 1.1

Abstract

The paper gives information helpful for system designers that want to use Alcatel's Web Touch One CPU board with LCD and touchpad in their own systems. Hardware as well as software parts information needed to adopt the board to other projects is provided there. This document doesn't pretend to be full technical documentation of the Alcatel device, but hope will fill some gaps in knowledge about this cool product.

Author:
Sebastian
SebaX01@op.pl

Release	Version	Info
16-04-2004	1.0	Initial release
05-06-2004	1.1	Added information about programming flash. FAQ added.

Contents:

SYSTEM INFORMATION....3

MAIN COMPONENTS LIST....4

POWER SUPPLY....5

MAKING THE BOARD RUNNING....6

BACKLIGHT....6
LCD AND CPU....6
LISTENING DEBUGS....7
ANALOGUE PART....8
TOUCHPAD (ADC)....8

PROGRAMMING THE BOARD....9

ADDRESS SPACE....9
NAND FLASH....9
PROGRAMMING FLASH....10
JTAG/BDM interface....10
Burning new software....10
Short U-Boot course....11

APPENDIX 1 ALCATEL STARTUP DEBUGS....12

APPENDIX 2 USEFUL LINKS....15

APPENDIX 3 – GDB MESSAGES....16

APPENDIX 4 – FAQ....18

System information

During writing the documentation there was no entire phone available for the author but only CPU board, LCD with touchpad and backlight power supply. That is reason why some information are incomplete, especially about signals provided from externals boards.

CPU board is populated with:

- PowerPC processor running on 66MHz (XPC823ZT66A),
- 32MB of SDRAM configured in two 32-bit banks,
- 2MB of flash memory configured in 32-bit,
- 8MB of NAND flash memory,
- 33.6kbit/s Rockwell modem,
- analogue parts used mainly for voice processing.

LCD is tied with CPU board through 22-pin connector; touch panel located on LCD joins with CPU board by 4-pin connector. For board programming JTAG/BDM connector may be used. Detailed connector description is provided in chapter “programming the board”. Below pictures shows both sides of the CPU board. All chips were numbered to provide easy reference in documentation.

Fig. 1 CPU PCB top layer

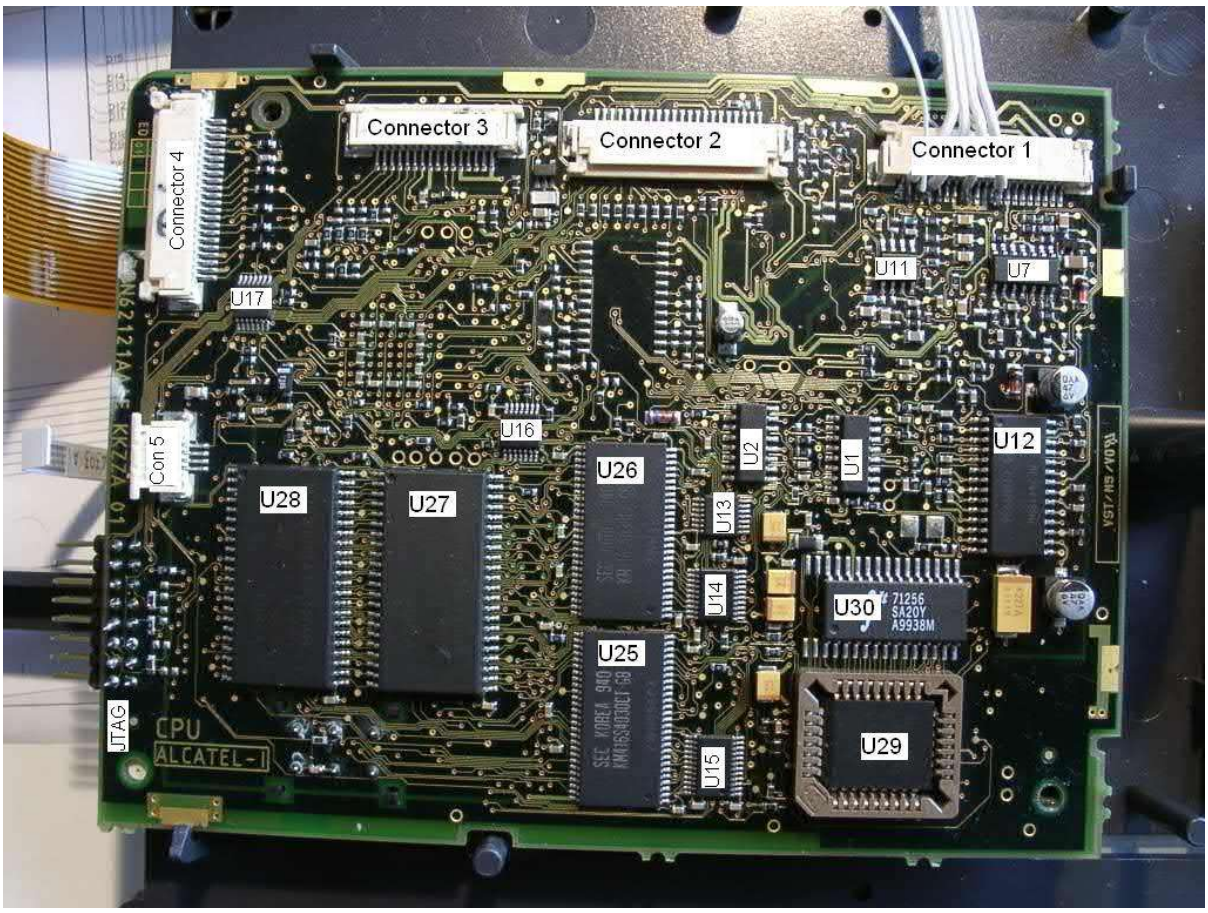

Fig. 2 CPU PCB bottom layer

Main components list

#	Component name	Component function
U1	74HC4053	Four two-input analogue switch
U2	74HC4053	Four two-input analogue switch
U3	74HC4053	Four two-input analogue switch
U4	74HC4051	Two four-input analogue switch
U5	74HC4053	Four two-input analogue switch
U6	MC3403	Four operational amplifiers
U7	MC3403	Four operational amplifiers
U8	MC34119	Audio low power amplifier
U9	MC3403	Four operational amplifiers
U10	74HC574	Eight-bit latch
U11	MC34072	Two operational amplifiers
U12	MC34118DW	Headset audio circuit
U13	74LVC14	Six NOT gates
U14	74HC574	Eight-bit latch
U15	74LVC4245	Eight-bit bidirectional gate with logic level shifting
U16	74LC32A	Four OR gates
U17	74LVC08A	Four AND gates
U18	74HCT541	Eight-bit buffer
U19	MC14LC5480	Audio codec
U20	LM324	Four operational amplifiers

U21	XPC823ZT66A	PowerPC processor MPC823 series
U22	KM29U64000T	8MB NAND Flash
U23	KM416S4030CT	4MB*16 SDRAM
U24	KM416S4030CT	4MB*16 SDRAM
U25	KM416S4030CT	4MB*16 SDRAM
U26	KM416S4030CT	4MB*16 SDRAM
U27	M29W800AT	512KB*16 Flash
U28	M29W800AT	512KB*16 Flash
U29	M27C1001	128KB*8 OTP EPROM
U30	IDT71256	32KB*8 CMOS Static RAM
U31	RCV336ACFW	33,6 kbps modem

Power supply

Because no original power supply was provided with the module, below information in some areas cannot be confirmed. Especially voltages for analogue part of the module are guess only.

All voltages are provided through connector 1 (the one close to 8-pin MC34072 chip):

pin	**voltage**	**notes**
1	-5V??	-5V?? - for analogue circuits
2	+12V??	+12?? - for speaker amplifier
3	0V??	0V?? - for speaker amplifier
4	+5V	+5V- for analogue circuits
5,6,7	0V - GND	analogue GND
8..10	+5V	+5V supply for logic circuits
11..17	0V - GND	digital GND
18,19	+3.3V	+3.3V supply for logic circuits
20	MAGIC	That’s magic pin. You have to bangle it to start original software
21	0..3V3	PWM output from processor. Probably to slightly control 36.6V
22	+36.6	LCD drive power supply (regulated +/- 1V to achieve best contrast)

Making the board running

This chapter describes consecutive steps to make the board running.

Backlight

The easiest part is to make backlight working. The small separate board with 2-pin connector is a power converter for it. Unfortunately because no original PSU was available it cannot be said what voltage should be applied to this connector. For sure the backlight will lit (but how long?) when +10V power is provided to this connector. Where the plus and where the minus should be applied can be recognized thanks to descriptions on capacitor placed close to connector.

LCD and CPU

After that it would be nice if some pictures appeared on LCD. This is a bit harder and requires power supply with three voltages. The logic circuits must be working – connect +5V to pins 8 to 10 of connector 1 and +3.3V to pins 18 and 19 of the same connector. To make LCD working +36.6V to pin 22 must be provided. Don't forget to connect common GND the module (pins 11..17). After turning on above voltages nothing happens – the software located in Flash waits for something. To kick it you have to bangle pin 20 of connector 1. It is suggested to connect the pin to + 5V through 1k resistor and solder switch button to GND. Now power the module on, wait few seconds, press the button and welcome screen will appear.

Fig. 3 Welcome screen

Now there is possibility to correct contrast changing +36.6V a bit. After longer wait it can be seen application screen.

Fig. 4 Application screen

Playing with the huge switch button located under LCD will effect in other pictures shown on screen.

Listening debugs

On the JTAG connector there is serial port attached also. Module with original software, during startup, prints there debugs. The signal voltage level is 3.3V thus voltage converter is required to connect module with PC RS232 port. Any of widely available RS232 converters can be used there, preferably with 5V supply, to do levels adjustment. As there is also +5V supply output on the JTAG connector, making such conversion should not be a problem. Finally after wiring the module with PC startup debugs will appear on terminal.

```
CHORUS/ClassiX r3.1 for MPC860ADS
Copyright 1998 Sun Microsystems, Inc. All rights reserved.

Kernel modules : CORE SCHED_CLASS [ FIFO RR TS RT ] SEM MIPC MEM_FLM TICK
KBIM ENV ETIMER LOG MUTEX EVENT DATE TIMEOUT LAPBIND
IOM: Failed to connect to KDB
PPP: ppp0 ifnet interface available
PPP: ppp1 ifnet interface available
IOM warning: cannot attach sl0 or ndm0
IOM Internet address: 0.0.0.0  netmask: 0x0
```

```
Actinit starting (console messages available)...
```

All debugs are presented in Appendix 1.

Above information appears after turning on the module. The transmission is 9600 bps, 8N1. Of course no hardware flow control. The software doesn't reply to any character sent to it – only watching debugs is possible.

Analogue part

In order to make analogue part working additional voltages have to be provided to the board. Looking into connector 1 description it can be found where and what voltages should be applied. No tests were done with that, so nothing can be guaranteed. Before implementing own software hard work on this area seems not to be justified.

Touchpad (ADC)

Unfortunately no touchpad PCB was provided with the module. All information there is based on "educated guess" only, so some mistakes may appear. The Analogue-Digital Converter that serves touchpad is tied through connector 3. Together with analogue part of touchpad circuit located on CPU PCB it allows finding where user put his finger. So, without this board it is impossible to make touchpad working. On the board ADC chip is located that converts analogue signals coming from touchpad circuit and from LCD thermometer into digital domain and sends that values through SPI. Probably chip with functionality similar to MAX192 was used. Designing board which replace original one is not really complicated – please read Motorola specifications concerning touchpad service available in MPC821TPDDRV.zip archive on theirs website.

Programming the board

This entire documentation is written with purpose. No, I don't want to provide the phone service manual – not at all. My goal is to use the board in my house as a control center. The main processor has much enough MIPS power to do such a job. Large LCD with touchpad is giving me freedom from attaching keyboard to the device. There is only one missing point: Ethernet. When I finish debugging the board I am going to attach Ethernet PHY to the processor or connect external Ethernet controller.

Somebody may say that writing software for such a complicated system is a task for huge group of engineers. I would agree with that if there was no Linux. Thanks to many contributors implementing the system on different processors we can think on porting it to our board. As the board main processor is PowerPC, supported by Linux, it is possible to use it on this platform. There is very interesting project called "Das U-Boot" that works well on many PowerPC (and not only) platforms and is based on the penguin system. This is kernel bootlader that is able to load Linux system from different locations (flash, Disk On Chip, HDD, Ethernet, RS232). Because the Alcatel board doesn't have Ethernet controller (so far, so far) the large Nand Flash is best place to store the kernel that will be run by "Das U-Boot".

Address space

For proper board initialization and usage, configuration of system components must be known. The entire system address space is divided to banks (read MPC823 datasheet for details) in which different devices are located. Below table describes them:

Bank #	Device	Notes
0	2*29W800T - flash with system code	2MB of 32-bit flash
1	2*KM416S4030 SDRAM – system memory	first 16MB of 32-bit SDRAM
2	2*KM416S4030 SDRAM – system memory	second 16MB of 32-bit SDRAM
3	Not used	
4	KM29U64000T Nand Flash	8MB 8-bit Nand Flash
5	74LVC4245 bidi port with voltage conversion	gate to modem chip
6	2*74HC574 – 8-bit latch register	additional PIOs
7	Not used	

On some boards different chips may be placed e.g. MT48LC4M16A2 SDRAM, but configuration stays same.

NAND Flash

Access to Nand Flash is done through bank 4, but many control signals are coming directly from processor PIOs. Below table shows connections of Nand flash signals with processor PIOs:

PIO	Nand signal	function
PC9	/WP	write protect
PC13	ALE	Address Latch Enable. if high next write is an address writing
PC14	CLE	Code Latch Enable. If high then next write is a command write
PC15	/CE	Chip Enable. If low chip is selected
DP2	/BUSY	The busy signals goes through inverter to DP2 pin
---	/SE	Spare Enable. The signal is tied low

Signals /WE and /RE are generated dynamically using /CS4, /WE0 and /OE1 (GPLA1) signals.

Programming Flash

Using dedicated tool software may be downloaded using JTAG connector. As the signals are not connected in standard way below the connector pinout is provided:

PB24 (RxD1)	1	JTAG/BDM connector	2	PB25 (TxD1)
GND	3		4	N.C.
/SRESET	5		6	IPB0 (VFLS0)
TCK	7		8	GND
IPB1 (VFLS1)	9		10	GND
TDI	11		12	/HRESET
TDO	13		14	+5V
/TRST	15		16	TMS

JTAG/BDM interface

There is JTAG interface required to connect to processor and program flash. The MPCBDM hardware is cheap and good enough tool for doing the job. It is LPT-JTAG interface build using only two 74HC14 chips. Please visit mpcbdm homepage and study theirs BDM adapters that are supported by GDB. It is recommended to use version 2 as all init files are prepared for it.

Burning new software

When physical connection between Alcatel board and PC is established there is need for burning program that is able to connect with MPC823 and save data to onboard flash. Newest GDB has support for mpcbdm interface and together with configuration scripts will do the task. There are three files required for GDB: configuration script (mympc2.init), flash configuration file (flash.dev) and image to program the Alcatel board (u-boot.bin).
When GDB is run enter “iboard” command to configure board and then “progflash” that will burn the image in flash. After successful programming enter last command “mpcbdm reset” and below picture should appear on LCD:

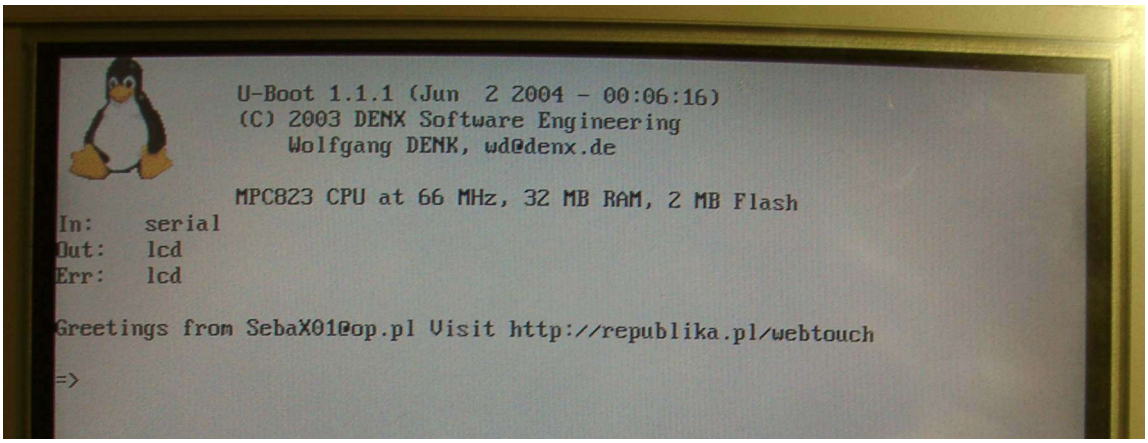
U-Boot 1.1.1 (Jun 2 2004 - 00:06:16)
(C) 2003 DENX Software Engineering
Wolfgang DENK, wd@denx.de
MPC823 CPU at 66 MHz, 32 MB RAM, 2 MB Flash
In: serial
Out: lcd
Err: lcd
Greetings from SebaX01@op.pl Visit http://republika.pl/webtouch
=>

Fig. 5 Das U-Boot welcome screen

Short U-Boot course

After booting, system will accept commands from UART that is located on JTAG connector. Transmission is done in 115kbit/s 8N1 format. Below are some commands that may be interesting:
“setenv stdout serial” – redirects output messages from LCD to UART
“bmp display <address> <x> <y>” – displays bitmap from memory on LCD screen
“loadb <address>” – loads binary file using Kermit protocol into memory (be sure that you redirected messages to UART before),
“go <address>” – runs program from memory (for example loaded before using loadb/loads commands”
and many others, please enter “? or help” for more info.

Das U-Boot loader allows loading new applications into system memory, and then running them. Thanks to this any required behavior (for example dancing penguins on screen) can be achieved. In uboot.tgz archive there are two applications included that can be used in early tests “hello_world” and “timer”.

Appendix 1 Alcatel startup debugs

```
CHORUS/ClassiX r3.1 for MPC860ADS
TOOLS APP REGI
Copyright 1998 Sun Microsystems, Inc. All rights reserved.S APP REGISTERED
TO MPCKEYBORD

Kernel modules : CORE SCHED_CLASS [ FIFO RR TS RT ] SEM MIPC MEM_FLM TICK
KBIM EAPP REGISTERED TO MPCKEYBORD
TOOLS APP REGISTE
NV ETIMER LOG MUTEX EVENT DATE TIMEOUT LAPBIND


IOM: Failed to connect to KDB
PPP: ppp0 ifnet interface available
PPP: ppp1 ifnet interface available
IOM warning: cannot attach sl0 or ndm0
IOM Internet address: 0.0.0.0  netmask: 0x0

Actinit starting (console messages available)...

DRVMISC: starting (console messages available)...
inet: Will get DNS addresses from environment variables DNS1 and DNS2
inet: inet name server daemon started
notice: CHORUS PPP Daemon started (release 4.0)
done
notice: Waiting for instructions (PPP multi line mode)
Personal Jazz - Running JDK 1.1 - Alcatel version 1.5
Alcatel modifications :
        javai.c         : no jazzDrvConnect
        color.c         : new CLUT
        properties_md.c : try to use JAVA_HOME
        classloader.c   : performances tags
JAZZ: using SDRAM_SIZE=32768
JAZZ: MaxHeapSize set to 25165824
PANIC_RESERVE_RED set to 150000
PANIC_RESERVE_YELLOW set to 300000
MAX_STREAM_NUMBER set to 48
HANDLE_RESERVE_YELLOW set to 2000
HANDLE_RESERVE_RED set to 500
NATIVE_RESERVE_YELLOW set to 200000
NATIVE_RESERVE_RED set to 100000
STREAM_RESERVE_YELLOW set to 8
STREAM_RESERVE_RED set to 4
THREAD_RESERVE_YELLOW set to 48
THREAD_RESERVE_RED set to 43
Java Heap size reset to 17825792 Native Heap size set to 7340032
FREE_MEMORY_LOW_WATER_MARK set to 20
Restricted heap: Size 17825792 bytes
InitializeMalloc: use 7340032 byte(s).
JaZZ: Failed to connect to KDB


pJaZZ r1.0

JavaOS Version: 1.1M

MPCFrameBuffer instanciated ...
MPCFrameBuffer.init(8, 8)
MPCFrameBuffer sets to 640x480
```

```
MPCFrameBuffer CLUT initilization
FrameBuffer address is 154247168
aw FrameBuffer initialization


ASAP version : 1.0-
mouseDevice  : TOUCHSCREEN
audio        : ON
scrolling    : simple

Can't create remote manager !
Keyboard thread is launched ...

* CAD * ASAP10
AWInputEventHandler launched
OIM-java.home is /datarom
Sat Jan 01 06:00:00 GMT+01:00 2000
WebPhoneTicker readjusted without tick notification
ProxyMgr : removing http proxy
ProxyMgr : removing https proxy
DEBUG : sun.net.www.protocol.https.HttpsClient implementation of https
Registering MPCMouse to StandByMode
MPCMouse registered to StandByMode
DEBUG : sun.net.www.protocol.https.HttpsClient implementation of https
DIS.Static:UserGlobals.
DIS.Static:UserGlobals.getPropertyt:Cookies=1
CAD> Mail: iso-8859-15 available
There is 1 dialogs on the screen

Setting Dialog
Disabling InputMethod
ajout d'un SmartcardListener
Trying to load drivers from: /datarom/jsc
Trying to load drivers from: //.jsc
ajout du driver WPADriver
thread is launched on webphone
STARTING NEW DIALOG THREAD
TOOLS APP REGISTERED TO MPCKEYBORD
TOOLS APP REGISTERED TO MPCKEYBORD
TOOLS APP REGISTERED TO MPCKEYBORD
TOOLS APP REGISTERED TO MPCKEYBORD
TOOLS APP REGISTERED TO MPCKEYBO
TOOLS APP REGISTERED TO MPCKEYBORD
TOOLS APP REGISTERED TO MPCKEYBORD
TOOLS APP REGISTERED TO MPCKEYBORD
TOOLS APP REGISTERED TO MPCKEYBORD
TOOLS APP REGISTERED TO MPCKEYBORD
TOOLS APP REGISTERED TO MPCKEYBORD

Initialize AudioDriver_ASAP10
ProductionController: will use 8 ImageFetcher threads
ProductionController: will use 8 ImageFetcher threads
mini PANIC browser.destroyAllApplets failed
java.lang.NullPointerException
        at sunw.hotjava.bean.HotJavaBrowserBean.destroyAllApplets()
done
info: Debug turned OFF (level=4)
        at sunw.hotjava.bean.HotJavaBrowserBean.destroyAllApplets()
        at com.alcatel.pwa.Instance.unregister()
        at com.alcatel.pwa.BridgeImpl.unregister()
        at wpa.gui.apps.BrowserApp.realUnregister()
```

```
StdWaitingPopup : tick : true
        at wpa.WPGlobals.openWithWaitingPopupMethod()
        at com.alcatel.jazz.MethodThread.run()
StdWaitingPopup : tick : false
StdWaitingPopup : tick : true
There is 0 dialogs on the screen
```

Appendix 2 Useful links

http://republika.pl/webtouch – this project homepage

http://www.motorola.com/ - PowerPC XPC823 page (search for MPC823)

ftp://ftp.leo.org/pub/eldk/ - PowerPC C compiler
ftp://ftp.denx.de/pub/u-boot/ - Das U-Boot releases
http://www.vas-gmbh.de/software/mpcbdm - MPCBDM interface homepage

Used chips datasheets:
http://e-www.motorola.com/files/if/cnb/MPC823EUM.pdf
http://wulfman.com/pdffiles/KM416S4030C.PDF
http://www.national.com/ds.cgi/LM/LM3303.pdf
http://decoding.newmail.ru/pdf/M29W800AB90N1.pdf
http://www.mcu-memory.com/datasheet/samsung/km29u64000(i)t.pdf

http://www.semiconductors.philips.com/pip/74HC574N.html
http://www.semiconductors.philips.com/pip/74LVC08AD.html
http://www.semiconductors.philips.com/pip/74LVC14AD.html
http://www.semiconductors.philips.com/pip/74LVC32AD.html
http://www.semiconductors.philips.com/pip/74HC541N.html

Appendix 3 – GDB messages

```
[root@Atlas MPCBDM]# ./start.sh
GNU gdb Yellow Dog Linux (5.2.1-4b_4)
Copyright 2002 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.  Type "show warranty" for details.
This GDB was configured as "--host=i386-redhat-linux --target=ppc-linux".
MPCBDM version 1.2.3 / 2001/10/25
got access rights for printer port 0 addr 0x378..0x37A
disabled power at port 0
adapter version 2 initialized
*** init
Target cpu PVR=0x00500000 PARTNUM=0x21 MASKNUM=0x00 REV_NUM=0x0000
warning: unknown CPU. Using default register description
BDM initialized
0xfff00100 in ?? ()
(gdb) iboard
Processor Initialized.
Memory initialized.
Ident Sequence: Ok!
(gdb) progflash
Erasing flash. Please wait...
checking FLASH [0x40100000,0x40100fff]...
checking FLASH [0x40101000,0x40101fff]...
checking FLASH [0x40102000,0x40102fff]...
checking FLASH [0x40103000,0x40103fff]...
checking FLASH [0x40104000,0x40104fff]...
checking FLASH [0x40105000,0x40105fff]...
checking FLASH [0x40106000,0x40106fff]...
checking FLASH [0x40107000,0x40107fff]...
checking FLASH [0x40108000,0x40108fff]...
checking FLASH [0x40109000,0x40109fff]...
checking FLASH [0x4010a000,0x4010afff]...
checking FLASH [0x4010b000,0x4010bfff]...
checking FLASH [0x4010c000,0x4010cfff]...
checking FLASH [0x4010d000,0x4010dfff]...
checking FLASH [0x4010e000,0x4010efff]...
checking FLASH [0x4010f000,0x4010ffff]...
checking FLASH [0x40110000,0x40110fff]...
checking FLASH [0x40111000,0x40111fff]...
checking FLASH [0x40112000,0x40112fff]...
checking FLASH [0x40113000,0x40113fff]...
checking FLASH [0x40114000,0x40114fff]...
```

```
checking FLASH [0x40115000,0x40115fff]...
checking FLASH [0x40116000,0x40116fff]...
checking FLASH [0x40117000,0x40117fff]...
checking FLASH [0x40118000,0x40118fff]...
checking FLASH [0x40119000,0x40119fff]...
checking FLASH [0x4011a000,0x4011afff]...
checking FLASH [0x4011b000,0x4011bfff]...
checking FLASH [0x4011c000,0x4011ccdf]...
Copying to Flash using host routines...
writing FLASH [0x40100000,0x40100fff]...
writing FLASH [0x40101000,0x40101fff]...
writing FLASH [0x40102000,0x40102fff]...
writing FLASH [0x40103000,0x40103fff]...
writing FLASH [0x40104000,0x40104fff]...
writing FLASH [0x40105000,0x40105fff]...
writing FLASH [0x40106000,0x40106fff]...
writing FLASH [0x40107000,0x40107fff]...
---Type <return> to continue, or q <return> to quit---
writing FLASH [0x40108000,0x40108fff]...
writing FLASH [0x40109000,0x40109fff]...
writing FLASH [0x4010a000,0x4010afff]...
writing FLASH [0x4010b000,0x4010bfff]...
writing FLASH [0x4010c000,0x4010cfff]...
writing FLASH [0x4010d000,0x4010dfff]...
writing FLASH [0x4010e000,0x4010efff]...
writing FLASH [0x4010f000,0x4010ffff]...
writing FLASH [0x40110000,0x40110fff]...
writing FLASH [0x40111000,0x40111fff]...
writing FLASH [0x40112000,0x40112fff]...
writing FLASH [0x40113000,0x40113fff]...
writing FLASH [0x40114000,0x40114fff]...
writing FLASH [0x40115000,0x40115fff]...
writing FLASH [0x40116000,0x40116fff]...
writing FLASH [0x40117000,0x40117fff]...
writing FLASH [0x40118000,0x40118fff]...
writing FLASH [0x40119000,0x40119fff]...
writing FLASH [0x4011a000,0x4011afff]...
writing FLASH [0x4011b000,0x4011bfff]...
writing FLASH [0x4011c000,0x4011ccdf]...
FLASH writing completed
(gdb) mp reset
(gdb)
```

Appendix 4 – FAQ

1. **Q: I have such a board but I don't know how I can use it.**

A: That board is complete processor system with LCD and touchpanel. I am sure that there are many applications that this board will fit. But you have to know that it requires large programming skills and many hours of code study. Think if it is worth for you.

2. **Q: I have the board but it seems to be broken. How to check it?**

A: Well, just go trough steps described in "Making the board running" chapter.

3. **Q: I did everything but LCD is black.**

A: The LCD should not be black. If it is that means your backlight doesn't work. Check if you supplied it correctly. Remember that nothing is visible on LCD until backlight is not working.

4. **Q: I did everything but only I can see is backlight.**

A: Check if there are messages on UART at least. If they are not – check your connections and PSU.

5. **Q: I can see messages on UART, but on LCD only backlight works.**

That means LCD is not enabled. Check pin 3 of U17 – there should be high level to enable LCD. If it is low try to connect 4k7 pull-up on pin 2 (yes, pin 2) of U17 to +3.3V.

6. **Q: Original Alcatel application works, but I can't download Das U-Boot.**

A: Check your MPCBDM interface. Check if GDB displays messages as shown in appendix 3. Try to define your problem more precisely.

7. **Q: GDB software cannot connect to board. Where can be a problem?**

A: There can be several reasons: wrong GDB version, interfaces wrongly build no read/write rights to LPT device.

8. **Q: It seems that GDB software connects to board, but I can't program flash.**

The flash.dev contains definitions for ST M29W800AT chips. Check if you have the same chips, if not – try to modify flash.dev file.

9. **Q: I programmed flash, but the system is not working. No LCD info, no messages on** UART.

A: There could be two reasons: broken hardware or different hardware. The u-boot was prepared for board that is described in this document. Check if all components you have are identical to those specified in this document.

10 Q: I am afraid to break my Alcatel by tampering with flash.

Does it mean your phone is working now? If yes I would not touch that nice internet phone, and spend time on internet chats. Anyway, you can save original application to file to restore it in future. Have a look at mympc2.init file, where adequate functions are implemented.

11 Q: The u-boot works great, I can load my application into RAM, but I loose it after power down. Is there a way to avoid it?

Have a look at “cp” function. It can copy your application into flash.

12 Q: I have copied the application into flash, I checked it is there, but when I am trying to run it system hangs. Why?

Your application probably expects to be run from RAM. You can copy it to RAM from flash just before starting, or you may modify it to avoid storing variables in flash area for example by modifying linker script.

13 Q: Where is RAM area and where is the flash in address space?

32MB RAM area starts from address 0x00000000 when flash is located at 0x40000000 offset. Please check “bdinfo” and “flinfo” functions to get more information. Try to avoid using end of RAM area, because there is u-boot relocated.

14 Q: I am trying to write application to the board, but software I wrote doesn’t work. What I am doing wrong?

Honestly, I don’t know, but try to check examples in Das U-Boot releases and try to understand it. Next modify them slightly and adapt for your needs.